SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2022, Vol. 65 182104:1–182104:16
https://doi.org/10.1007/s11432-021-3396-7

Fast target-aware learning for few-shot video object segmentation

Yadang CHEN[1], Chuanyan HAO[2], Zhi-Xin YANG[3*] & Enhua WU[4,5]

[1]*Engineering Research Center of Digital Forensics, Ministry of Education, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing* 210044*, China;*
[2]*School of Education Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing* 210023*, China;*
[3]*State Key Laboratory of Internet of Things for Smart City, Department of Electromechanical Engineering, University of Macau, Macao* 999078*, China;*
[4]*State Key Laboratory of Computer Science, Institute of Software, University of Chinese Academy of Sciences, Beijing* 100190*, China;*
[5]*Faculty of Science and Technology, University of Macau, Macao* 999078*, China*

Received 12 July 2021/Revised 20 October 2021/Accepted 3 December 2021/Published online 27 July 2022

Abstract Few-shot video object segmentation (FSVOS) aims to segment a specific object throughout a video sequence when only the first-frame annotation is given. In this study, we develop a fast target-aware learning approach for FSVOS, where the proposed approach adapts to new video sequences from its first-frame annotation through a lightweight procedure. The proposed network comprises two models. First, the meta knowledge model learns the general semantic features for the input video image and up-samples the coarse predicted mask to the original image size. Second, the target model adapts quickly from the limited support set. Concretely, during the online inference for testing the video, we first employ fast optimization techniques to train a powerful target model by minimizing the segmentation error in the first frame and then use it to predict the subsequent frames. During the offline training, we use a bilevel-optimization strategy to mimic the full testing procedure to train the meta knowledge model across multiple video sequences. The proposed method is trained only on an individual public video object segmentation (VOS) benchmark without additional training sets and compared favorably with state-of-the-art methods on DAVIS-2017, with a $\mathcal{J}\&\mathcal{F}$ overall score of 71.6%, and on YouTubeVOS-2018, with a $\mathcal{J}\&\mathcal{F}$ overall score of 75.4%. Meanwhile, a high inference speed of approximately 0.13 s per frame is maintained.

Keywords video object segmentation, few-shot, target-aware, meta knowledge, bilevel-optimization

Citation Chen Y D, Hao C Y, Yang Z-X, et al. Fast target-aware learning for few-shot video object segmentation. Sci China Inf Sci, 2022, 65(8): 182104, https://doi.org/10.1007/s11432-021-3396-7

1 Introduction

Video object segmentation (VOS) aims to separate foreground objects from the background in a video sequence and is a fundamental task in computer vision; it has important applications, such as video detection [1], classification [2], and reconstruction [3]. Typically, according to whether or not annotations are provided for the first frame during testing, VOS can be categorized into few-shot VOS [4, 5] and zero-shot VOS [6]. In this study, we focus on few-shot VOS, where the ground-truth segmentation mask of an object is given in the first frame of a video sequence. The task is then to accurately estimate the segmentation of the object for the rest of the video.

Advances in deep learning and the introduction of the DAVIS[1)] and YouTubeVOS[2)] challenge have led to significant progress in semi-supervised VOS. Aiming to be target-aware with a limited annotation, quite a few methods rely heavily on first-frame finetuning [4, 5, 7–9]. Although they have achieved high accuracy, they have high computational costs and are impractical for real-time cases. An alternative

* Corresponding author (email: zxyang@um.edu.mo)
1) https://davischallenge.org/index.html.
2) https://youtube-vos.org.

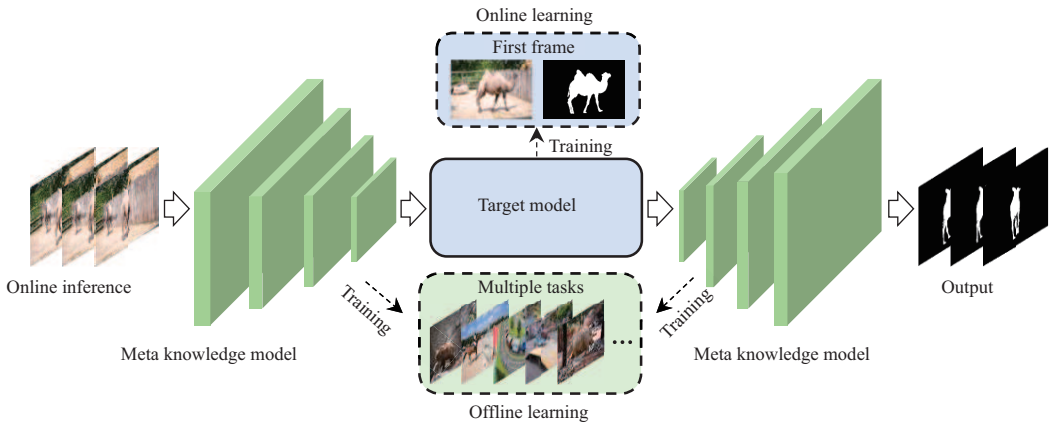

Figure 1 (Color online) Brief diagram of the proposed VOS approach. Given a test video sequence to be segmented, a powerful target-aware model is first learned by minimizing the segmentation error in the first frame. For the subsequent frames, we use the learned model to predict the segmentation mask. During the offline training stage, we use a bilevel-optimization strategy to mimic the full inference procedure to train the meta knowledge model across multiple segmentation tasks.

approach is to use the previous segmentation result to guide the prediction of the current frame, known as mask propagation [4, 10–13]. However, they are sensitive to occlusion, exposure, or fast motion during the propagation. Recent approaches address these limitations by employing a metric matching-based model [14–17], in which the final segmentation is predicted by pixel-wise matching in a learned metric space or called embedding space in their studies. Similarly, nonlocal self-attention matching [18] is also employed to find the correspondence of target objects between the current and past frames in quite a few state-of-the-art studies [19–24]. Although these memory matching-based methods achieve good accuracy, one drawback is the requirement for large amounts of data to train such a meticulous network. As a result, they usually rely heavily on complicated pretraining on large-scale image datasets, which are unsuitable for most practical applications.

To this end, we propose a fast target-aware learning approach for few-shot VOS, by which the system learns a powerful representation of the target and background appearance from a limited first-frame annotation. Specifically, we divide the proposed network into two models. First, the meta knowledge model initially learns the general semantic features for the input video image and then refines the coarse predicted mask to produce a fine result. Second, the target model adjusts the network according to the specific target from the first-frame annotation. During the online inference stage for each video segmentation task, we employ fast optimization techniques to train a powerful target model by minimizing the segmentation error in the first frame and then use it to predict the subsequent frames. During the offline training stage, we use a bilevel-optimization strategy to mimic the full inference procedure to train the meta knowledge model across multiple segmentation tasks. A brief illustration is shown in Figure 1. The proposed VOS approach can be naturally viewed as meta learning (Subsection 3.1), as it expediently learns a specific model for each new video task.

Thanks to the proposed method, our system is easy to train and requires no more pretraining on image datasets. Furthermore, the employment of fast optimization techniques without the need for backpropagation enables real-time video segmentation. Importantly, the proposed method segments multiple objects in a single forward pass. To address the issues of significant appearance changes, an efficient online adaptation mechanism is employed to further improve accuracy. We train our method on individual public VOS benchmarks without additional training sets and report the evaluation results on their validation sets. The proposed method is compared favorably with state-of-the-art methods, with a $\mathcal{J}\&\mathcal{F}$ overall score of 71.6% on DAVIS-2017 and with a $\mathcal{J}\&\mathcal{F}$ overall score of 75.4% on YouTubeVOS-2018. Without time-consuming finetuning, optical flows, or pre/postprocessing, our method maintains a high inference speed of approximately 0.13 s per frame.

2 Related work

Semi-supervised VOS is a common task in computer vision and is usually performed using graph theory [25–27]. With the recent development of deep learning, deep learning based algorithms have achieved

remarkable performances on semi-supervised VOS tasks. In addition, public VOS benchmarks, such as the DAVIS [28] and YouTubeVOS [29] datasets, have significantly influenced this development.

2.1 First-frame finetuning and mask propagation

In [5], OSVOS uses a fully convolutional network (FCN) pretrained on ImageNet and DAVIS datasets for the semantic segmentation task, and it is finetuned on the first-frame ground truth of the target video at test time. In [30], OSVOS is extended to OnAVOS by using an online adaptation mechanism to handle appearance changes. In [8], OSVOS-S adds semantic information from an instance segmentation network. Heavy finetuning on the first frame significantly improves accuracy. However, it has a high computational cost.

An alternative approach is to use VOS as a mask-refinement process, where the previous mask prediction is adapted to fit the target in the current frame by using a convolutional neural network, known as MaskTrack [4] or MaskPropagation [10, 12]; however, errors accumulate seriously over time. This approach is also extended by [11, 13], which incorporates motion information through optical flows as an additional cue to improve accuracy. However, these methods cannot handle temporal discontinuities resulting from outliers.

On the basis of the two thoughts, a number of complicated techniques have been used for further improvements: LucidTracker [9] employs an elaborate data augmentation mechanism, whereas DyeNet [31] optimizes the segmentation process in iterative steps, where each step combines a temporal propagation and a re-identification module. Additional preprocessing (e.g., frame selection [32] or a pretrained decoder [33]) and postprocessing (e.g., Markov random field (MRF) optimization [34] or similarity-based aggregation [35]) have also been used to improve the results. Furthermore, DTMN [36] uses a recurrent neural network to fuse the output of two deep networks. Moreover, PReMVOS [7] combines four networks with finetuning and a merging algorithm. Although these methods have yielded promising results, they are seriously complicated and time consuming.

2.2 Pixel-wise memory matching

Recent approaches address these limitations by employing metric matching [14–17,37], in which the final segmentation is predicted by pixel-wise matching in a learned embedding space. In [37], PML uses a metric space learned with a triplet loss, and VOS is performed by a k-nearest neighbor (KNN) retrieving process. It is considerably faster and more flexible, but the result is noisy, owing to the hard assignments that are involved. Importantly, KNN retrieval is not differentiable. Unlike PML, VideoMatch [14] uses a soft matching layer and directly optimizes the resulting segmentation instead of using a triplet loss, resulting in an end-to-end trainable architecture. However, due to appearance changes and limited temporal information, they still suffer from the mismatching problem. To ease this problem, FEELVOS [15] and AGSS [16] use local and global matchings to guide the final segmentation decision; meanwhile, CFBI [17] extends foreground matching by collaborative foreground/background integration matching. They all have shown that using additional matchings is beneficial for segmenting the current frame. Accordingly, quite a few state-of-the-art studies [19–24] use more frames for the segmentation task, in which nonlocal self-attention matching [18] is employed to find the correspondence of target objects between the current and possibly all the past memory frames.

Although these memory matching-based methods have achieved state-of-the-art performances by making full use of the information from previous frames, a drawback is a requirement for large amounts of data to train such an elaborate network. As a result, they usually rely heavily on complicated pretraining on large-scale image datasets, which are unsuitable for most practical applications.

2.3 Meta learning for few-shot VOS

Most DNN-based methods rely on the ability to learn from large-scale annotations. In this study, the proposed learning approach is intended to expediently learn new tasks from limited information. One feasible strategy for few-shot learning is the notion of meta learning [38], which, however, has not been sufficiently explored in the context of video segmentation. Meta learning is most commonly understood as "learning to learn", whereby the network learns to update the segmentation model rapidly. A few recent attempts follow this direction, in which the behavior of the deep model is manipulated by batch normalization parameters [39], channel-wise attention mechanisms [16, 17] or graph optimization [40] conditioned on the

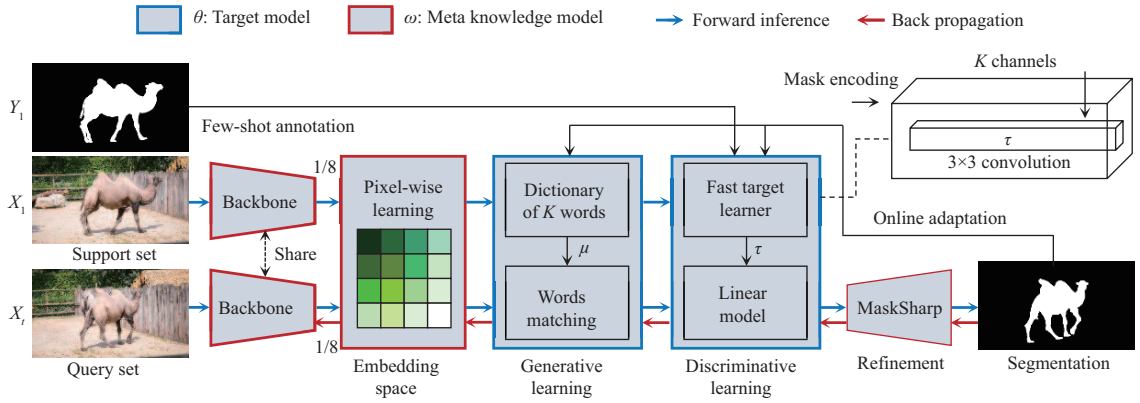

Figure 2 (Color online) Overview of the proposed semi-supervised VOS architecture. In this approach, meta knowledge is represented as a backbone extraction with an embedding head and a refinement network. The target model consists of a generative and a discriminative learning procedure.

first-frame annotation input. However, these deep networks are usually designed meticulously with heavy weights, resulting in an elaborate training procedure. Although some methods employ an approximate solution [41, 42] to optimize the discriminative network, they lead to computations that are intractable when aiming for acceptable speed, requiring extensive matrix multiplications. Recent generative methods attract our attention. They are based on, for example, K-means [43], Gaussian mixture model [12]. Such generative models have the advantage of facilitating efficient closed-form solutions that are easily integrated into neural networks, but they have a weaker capacity for target learning than discriminative networks.

In this study, we tackle the above problems by integrating generative and discriminative models to learn the target-aware appearance. Compared with recent methods, the proposed approach is quite flexible and lightweight. Moreover, it segments multiple objects in a single forward pass and can be easily trained using a bilevel-optimization strategy.

3 Method

We propose a fast target-aware learning approach for few-shot VOS that comprises two models: meta knowledge model ω and target model θ. An overview of the model is shown in Figure 2. For the meta knowledge model, we use an FCN module as the backbone to extract semantic features for each pixel, followed by an embedding layer, where the extracted pixel features can be embedded in a new space. Furthermore, MaskSharp network [44] is employed to up-sample a coarse segmentation mask to the original image size. For the target model, we integrate generative and discriminative learning for achieving a good balance of speed and strength as the aforementioned. Concretely, the target model encodes the image features by using a set of deep visual words and decodes them later by a compact linear model.

During the online inference for each video segmentation task, we employ fast optimization techniques to train a powerful target model by minimizing the segmentation error in the first frame. Then, we use them to predict the subsequent frames. During offline learning, the meta knowledge model is trained using a bilevel-optimization strategy across multiple segmentation tasks, which are often used in meta training [38]. That is, for each iteration, the meta knowledge model is alternately trained with the learned target model, and the target model is predicted with the learned meta knowledge model. To handle the problem of appearance changes, online adaptation mechanisms are also used to further improve accuracy using previously predicted results.

In the following, we first formulate the proposed VOS approach as a meta learning problem in Subsection 3.1. We then detail our target model in generative (Subsection 3.2) and discriminative (Subsection 3.3) learning. The online adaptation mechanism is described in Subsection 3.4. Finally, implementation details, including inference and training, are presented in Subsection 3.5.

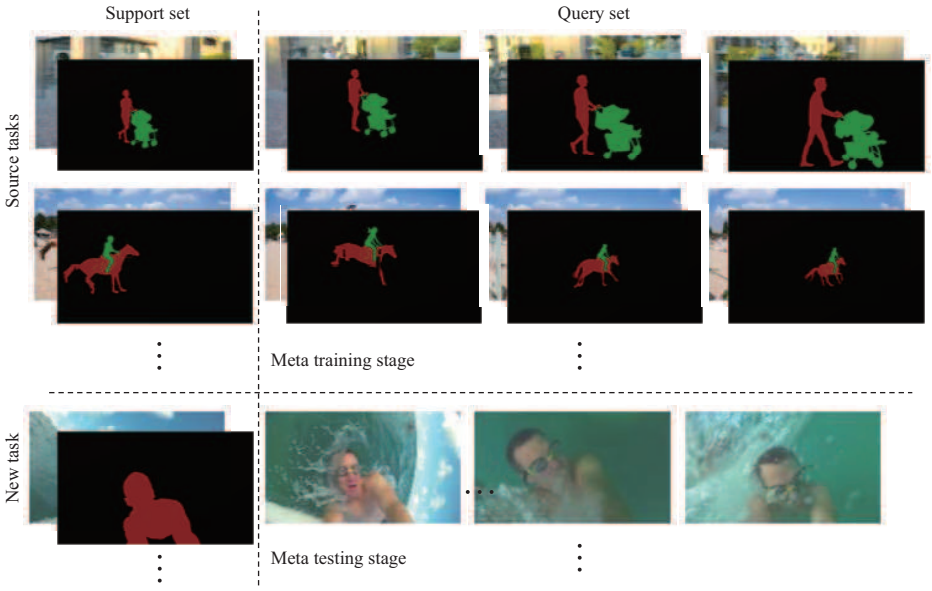

Figure 3 (Color online) Formulation of VOS as a meta learning problem. Each video sequence represents a task, including a first-frame annotation (support set) and the remaining frames (query set) to be segmented. In meta training, the objective is to learn across-task knowledge, which enables to learn a target-aware model in meta testing.

3.1 Problem definition

Meta learning or learning to learn is often used to understand a general-purpose learning algorithm that can generalize across tasks and ideally enable new tasks to be learned more effectively than previous tasks. The training and inference procedures in meta learning are often termed meta training and meta testing, respectively.

In this study, the objective of meta training is to learn model parameters ω on a variety of tasks (e.g., video sequences), which are sampled over distribution of tasks $p(\mathcal{T})$. Let $\mathcal{T}_{\text{Source}}$ denote the set of these sampled tasks. As each task has training (support) and validation (query) data, let $\{\mathcal{D}_{\text{Source}}^{\text{Support}} \cup \mathcal{D}_{\text{Source}}^{\text{Query}}\}^i \in \mathcal{T}_{\text{Source}}$ denote the ith task for meta training. We now formulate the objective as follows:

$$\omega^* = \arg\min_{\omega} \sum_{i \sim p(\mathcal{T})}^{\mathcal{T}_{\text{Source}}} \mathcal{L}_{\text{meta}} \left(\omega | \{\mathcal{D}_{\text{Source}}^{\text{Support}} \cup \mathcal{D}_{\text{Source}}^{\text{Query}}\}^i \right), \tag{1}$$

where $\mathcal{L}_{\text{meta}}(\omega | \{\mathcal{D}_{\text{Source}}^{\text{Support}} \cup \mathcal{D}_{\text{Source}}^{\text{Query}}\}^i)$ measures the loss of a model by using ω on the ith task with known labels, where ω is often referred to as across-task knowledge or meta knowledge.

Similarly, each new task has support and query sets, and let $\{\mathcal{D}_{\text{New}}^{\text{Support}} \cup \mathcal{D}_{\text{New}}^{\text{Query}}\}^j \in \mathcal{T}_{\text{New}}$ denote the jth new task during the testing stage. In meta testing, the learned meta knowledge is used to train the target model on only the $\mathcal{D}_{\text{New}}^{\text{Support}}$ of each previously unseen new task. It is formulated as

$$\theta_j^* = \arg\min_{\theta_j} \mathcal{L}_{\text{tar}} \left(\theta_j | \omega^*, \{\mathcal{D}_{\text{New}}^{\text{Support}}\}^j \right), \tag{2}$$

where θ_j^* results in a target-aware model for a specific task; it benefits from meta knowledge about the algorithm to be used.

Using these definitions, Figure 3 illustrates the definition of our work as a meta-learning problem. Various choices exist for meta representation, such as parameter initialization, hyperparameters, and optimizers. In this study, ω represents an embedding that enables to train a specific target model θ on a few-shot annotation for a new task (refer to Figure 2).

3.2 Generative learning

Our target model includes generative and discriminative learning procedures. Given the learned ω, we first use a generative appearance model to represent image content. The key difference between [12, 43] and ours is that the proposed model does not require prior object-class knowledge. Let $\mathcal{F}(x|\omega)$ denote

the embedded feature of pixel x, and for each task, the pixel $\hat{x}$ from the support set (e.g., the first frame) is used to construct a dictionary of visual words by the K-means algorithm, obtaining K clusters $C_1, \ldots, C_k, \ldots, C_K$ with

$$C_1^*, \ldots, C_K^* = \arg \min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \sum_{\hat{x} \in C_k} \|\mathcal{F}(\hat{x}|\omega) - \mu^k\|^2, \tag{3}$$

and the respective centroid is computed as

$$\mu^k = \frac{1}{|C_k^*|} \sum_{x \in C_k^*} \mathcal{F}(\hat{x}|\omega). \tag{4}$$

Therefore, let $\mu = \{\mu^1, \ldots, \mu^K\}$ denote the dictionary related to the support set. Note that any clustering algorithm (e.g., GMM) can be used here; the K-means algorithm is selected because it is computationally efficient and simple.

Given any pixel x now, the probability of assigning it to the kth word is directly computed as a posterior probability, termed "word matching", as follows:

$$p(C_k|x) = \frac{\exp(\cos(\mathcal{F}(x|\omega), \mu^k))}{\sum_{\mu^\star \in \mu} \exp(\cos(\mathcal{F}(x|\omega), \mu^\star))}, \tag{5}$$

where cos measures the cosine distance. Finally, all the pixels on frame X_t are computed using (5) for each word, which forms a mask encoding map M_t with K channels, and t is the time index through the video sequence. Although the mask encoding map is object-class-unknown, in principle, each component k of pixels belongs to only one object, and it provides a highly discriminative cue. In practice, directly using $\exp(\cos(\mathcal{F}(x|\omega), \mu^k))$ without a constant factor is beneficial. It can be interpreted as a component score that encodes an object assignment.

3.3 Discriminative learning

Accordingly, we detail our discriminative model that can be efficiently updated by minimizing the segmentation error using the first-frame ground truth. To this end, we employ a lightweight linear model τ and $\tau \in \mathbb{R}^{S \times S \times K \times C}$, which constitutes the weight of a convolutional layer with kernel size S. Unless otherwise specified, $S = 3$ in our study. K denotes the number of channels in M_t, and C denotes the count of objects, which is a specific number for each task. Although it is a complex model with a large capacity, it is also prone to overfitting and is computationally costly to learn.

Fundamental to our approach, discriminative learning parameters τ must be learned with minimal computational impact. To enable the deployment of fast converging optimization techniques, we adopt an L2 loss between the output of the target model and the ground-truth labels, weighted by an element-wise weight map. Given the support set that usually contains only one frame with its label, in the form of pair (X_t, Y_t), the loss measure appearing in (2) is defined as

$$\mathcal{L}_{\text{tar}} = \frac{1}{2N} \parallel W_t \cdot (M_t * \tau - Y_t) \parallel^2 + \frac{\lambda}{2} \parallel \tau \parallel^2, \tag{6}$$

where '$*$' denotes the convolution operation, N is the number of pixels on X_t, and the scalar λ controls the regularization term. To balance the impacts of target and background pixels, we employ the similar definition used in [41] to ensure that the target influence is not too small relative to the usually much larger background region. The weight map W_t is defined element-wisely as

$$w_t(i) = \begin{cases} \hat{n_t}/n_t, & y_t(i) = 1, \\ (1 - \hat{n_t})/(1 - n_t), & y_t(i) = 0, \end{cases} \tag{7}$$

where $\hat{n_t} = \max(n_{\min}, n_t)$, $n_t = N^{-1} \sum_i^N y_t(i)$, and i is the spatial index in the total N pixels on mask Y_t. We set $n_{\min} = 0.1$ in our approach.

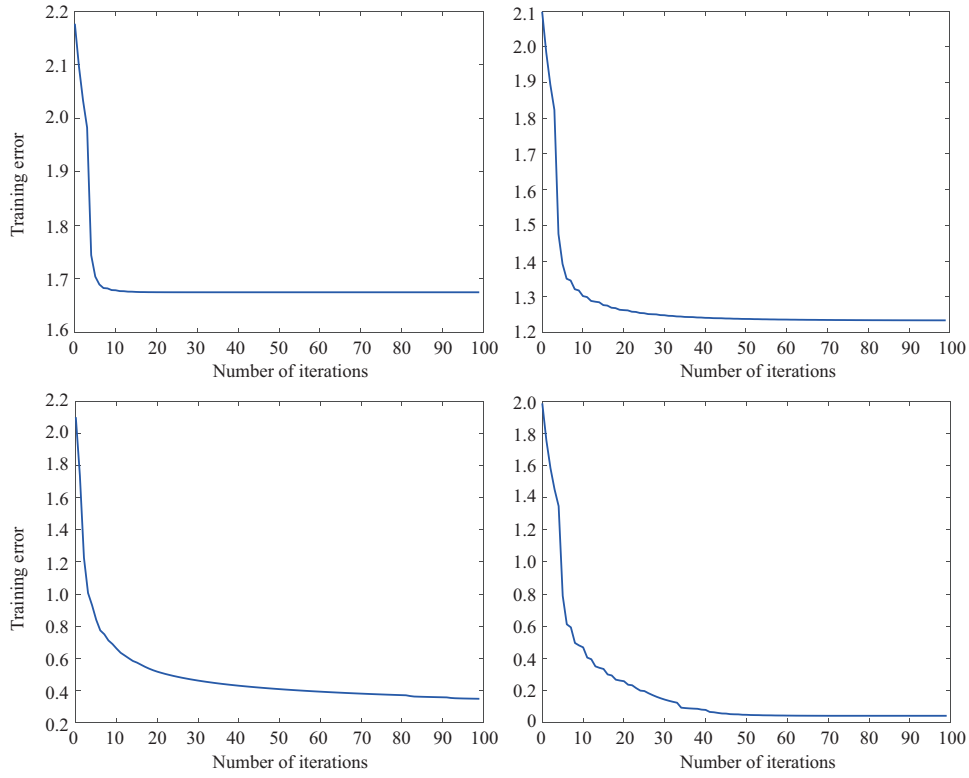

Figure 4 (Color online) Convergence examples of using the steepest descent strategy where the above four cases are with different meta knowledge ω. The target model can be optimized well enough in a few iterations.

Next, we need to choose an efficient approach to minimize (6), given the known ω and μ. To this end, we employ the steepest descent strategy to optimize parameters τ. For each iteration, the optimization can be expressed as $\tau = \tau - \alpha g$, where the gradient

$$g = \frac{1}{N} M_t *^{\mathrm{T}} (W_t^{\cdot 2} \cdot (M_t * \tau - Y_t)) + \lambda \tau, \tag{8}$$

and the step length

$$\alpha = \frac{\| g \|^2}{\frac{1}{N} \| W_t \cdot (M_t * g) \|^2 + \lambda \| g \|^2}, \tag{9}$$

$*^{\mathrm{T}}$ denotes the transposed convolution operation, and $\cdot 2$ is an element-wise square operation. In comparison to the common gradient descent approach, this strategy has significantly fast convergence properties that require only a few iterations, as displayed in Figure 4. The predicted τ is then used to segment the subsequent query frames. As clearly demonstrated in Section 4, the proposed target model learns to output powerful activations, leading to improved segmentation performances. Importantly, it makes our method easily segment multiple objects in a single forward pass.

3.4 Online adaptation

Objects often undergo occlusions, exposures, appearance changes, and other deformations through a video sequence. Consequently, adapting the segmentation model is necessary to achieve satisfactory performance. In this study, online adaptation is only performed on target model θ. That is, μ_t and τ_t learned from the pair (X_t, Y_t) are updated by the predicted result of $(X_{t+\delta}, Y_{t+\delta})$, where δ denotes a time interval (in terms of frames). We first use the K-means algorithm on $X_{t+\delta}$, initialized with the current estimate, that is μ_t, to compute an updated dictionary $\mu_{t+\delta} = \{\mu_{t+\delta}^1, \ldots, \mu_{t+\delta}^K\}$ using (4). Assuming that in the δ interval, the objects change slowly, and their pixel-level embeddings do not vary greatly, we replace μ_t^k with the new word $\mu_{t+\delta}^k$ only when $\cos(\mu_t^k, \mu_{t+\delta}^k) > \alpha$. Subsequently, we use the updated dictionary to recompute the mask encoding on frame $X_{t+\delta}$ using (5) and finetune τ_t by minimizing errors using the predicted result $Y_{t+\delta}$, also starting from the current estimate. To ensure that only reliable and

confident pixel-level predictions are involved in optimization, we apply a straightforward removal process by discarding pixels from the minimizing process if the predicted probability of the assigned label is under a threshold β.

3.5 Implementation details

In this study, we use DeepLab.v2 [45] as the backbone to extract features with an overall stride of 8. This model is based on Resnet-101 [46] with dilated convolutions. The backbone is followed by a depth-wise separable convolution layer to extract a pixel-wise embedding at the same stride, that is, a 3×3 convolution performed separately for each channel, followed by 1×1 convolution with 128 kernels. These 128-dimensional embeddings are then passed to the target model. Unless otherwise specified, the hyperparameters used in the target model are set as $K = 300$, $\delta = 5$, $\alpha = 0.2$, and $\beta = 0.6$. The influence of these parameters is analyzed in the ablation study in Subsection 4.1.

Inference. Given a new task $\{\mathcal{D}^{\text{Support}}_{\text{New}} \cup \mathcal{D}^{\text{Query}}_{\text{New}}\}$ during testing, the learned meta knowledge ω is used to predict the target model θ on only $\mathcal{D}^{\text{Support}}_{\text{New}}$, in which $\mathcal{D}^{\text{Support}}_{\text{New}}$ usually comprises the first frame X_1 with its ground truth Y_1. Specifically, the pixel-wise embedding of X_1 is first clustered into words forming μ, and the mask encoding is computed using (5) without a constant factor, as mentioned previously. Next, the discriminative learning predicts the linear model τ by minimizing the loss in (6) using Y_1. Note that τ is first initialized with zeros. Owing to the rapid convergence of the steepest descent strategy, only a few iterations ($N_{\text{init}} = 20$) are required to obtain a relatively good model. The learned dictionary μ and linear model τ are then applied to the subsequent test frame $X_t \in \mathcal{D}^{\text{Query}}_{\text{New}}$ to obtain the segmentation result. To handle scene changes, the target model is further updated using the predicted information from the processed frame. Specifically, the number of iterations is set to $N_{\text{update}} = 5$ for each update of τ in the subsequent frame, starting with the current estimate.

Training. A meta-training method is used for learning the meta knowledge ω defined in (1), where the training data consist of a number of different source tasks $\{\mathcal{D}^{\text{Support}}_{\text{Source}} \cup \mathcal{D}^{\text{Query}}_{\text{Source}}\}^i \in \mathcal{T}_{\text{Source}}$. Following the "episodic training" procedure in [47], we use only one task for each iteration. To mimic the full inference procedure, each iteration includes the following steps: (1) learning the target model θ on the support set with the fixed ω, (2) predicting the label of the query set using the learned model, (3) updating the meta knowledge ω by minimizing the error between the predictions and the ground truth of query set. Therefore, Eq. (1) can be viewed as a bilevel-optimization strategy for each source task:

$$\theta_i^* = \arg\min_{\theta_i} \mathcal{L}_{\text{tar}} \left(\theta_i | \omega, \{\mathcal{D}^{\text{Support}}_{\text{Source}}\}^i\right), \tag{10}$$

and

$$\omega^* = \arg\min_{\omega} \mathcal{L}_{\text{meta}} \left(\omega | \theta_i^*, \{\mathcal{D}^{\text{Query}}_{\text{Source}}\}^i\right), \tag{11}$$

where we design $\mathcal{L}_{\text{meta}}$ as the standard pixel-wise cross-entropy loss. Thus, we achieve learning to expediently learn a target-aware model from the first frame of the video over a pool of tasks. Concretely, each source task is constructed by randomly sampling a video from the training dataset, treating the first frame of the video as a support set, and randomly selecting three query frames from the rest of the video and treating them as a query set. However, the online adaptation mechanism is not simulated during training because the updating process is totally ω-irrelevant.

The training process begins by using weights of DeepLab.v2 [45]. The network is optimized using the Adam optimizer with the default momentum $(0.9, 0.999)$ for betas. The weight of the embedding layer is initialized using [48]. The model is first trained for 50k iterations at a learning rate of 10^{-3}, with the backbone weights fixed at half-resolution images. The complete network, including the backbone feature extractor, is then trained for 70k iterations at a learning rate of 10^{-4} at full resolution, followed by another training round (20k iterations) at a learning rate of 10^{-5}.

4 Experiment results

We evaluate our method on the public benchmarks DAVIS-2017 [28] and YouTubeVOS-2018 [29]. The evaluation metrics are the $\mathcal{J}$ score (i.e., average intersection-over-union ratio), the $\mathcal{F}$ score (i.e., average boundary similarity), and their mean value ($\mathcal{J}\&\mathcal{F}$ score). For DAVIS-2017, the scores are computed

Table 1 Ablative analysis of our approach on the DAVIS-2016 validation set

	Embedding	Generative learning	Discriminative learning	Training strategy	$\mathcal{J}\&\mathcal{F}$ (%)
REH	–	✓	✓	Bilevel-optimization	60.4
RTM	✓	–	–	Conventional training	54.9
RGL	✓	–	✓	Bilevel-optimization	67.1
RDL	✓	✓	–	Bilevel-optimization	65.4
Proposed architecture	✓	✓	✓	Bilevel-optimization	70.7

using the official metric code available on the DAVIS website[3)]. For YouTubeVOS-2018, we upload our results to the online server[4)] to obtain the scores. The method is trained and tested on a workstation with two Titan RTX GPUs and an Intel i9-8950HK CPU with six cores.

4.1 Ablation study

Here, ablation analysis is conducted for the key components of the proposed VOS method. We perform four alterations to the proposed architecture to evaluate the effect of each of its components.

(1) Removing embedding head (REH). Without the embedding head, the pixel-wise backbone feature is directly input into the appearance module for dictionary construction. In this variant, we freeze the backbone weights; thus, only the refinement network is trained across tasks.

(2) Removing target model (RTM). We remove the target model θ, including the generative and discriminative learning. To obtain an end-to-end trainable network, we adopt the standard binary pixel-wise cross-entropy loss as the prediction layer (i.e., 128-dimensional 1×1 convolution). This network is the same as the "parent network" introduced in OSVOS [5]. Thus, it can be trained in a conventional end-to-end manner without online target-aware finetuning.

(3) Removing generative learning (RGL). We remove the generative learning; thus, the mask encoding of each pixel becomes its embedding. This architecture is similar to RTM; the difference is that the weight of the final prediction layer is trained by a bilevel-optimization strategy using L_{tar} or $\mathcal{L}_{\text{meta}}$ loss alternatively, rather than the conventional end-to-end learning. This alteration is aimed at evaluating the effect of the bilevel-optimization training strategy used in this study.

(4) Removing discriminative learning (RDL). Removing the discriminative learning results in a method similar to that proposed in [12,43], which constructs an object-conditional dictionary for each object class according to the ground truth. Specifically, we set each foreground object as $K = 50$ words and the background as $K = 200$ words, following the previous study. Therefore, word matching is performed for each class dictionary. The final result is then aggregated using the probabilities computed from all matchings.

For simplicity, we train each architecture, including ours, with fixed backbone weights at a learning rate of 10^{-3} for 20k iterations. Given that the RTM variant cannot handle multiple object cases, this comparison is trained and evaluated on the DAVIS-2016[5)] without online adaptation. The results are shown in Table 1. The REH variant leads to a major reduction in the overall performance. This finding clearly demonstrates that the embedding layer is an essential component of the proposed VOS approach because the embedding pushes the pixels from the same object parts in the DAVIS dataset close to each other and pulls pixels from different parts far apart. The results also indicate that target-aware learning improves significantly even though RTM and RGL have a similar architecture. The weight of the final prediction layer is learned across the entire training set in RTM, whereas RGL predicts it through a lightweight online learning process, which is easily achieved through the bilevel-optimization strategy. Integrated generative and discriminative learning is also important. Specifically, the proposed method allows the network to represent an object by itself through learning, thus producing a more powerful target model than the object-conditional model used by RDL.

Dictionary size selection. In addition, we study the effect of dictionary size selection. The results are presented in Figure 5. Fixed learned meta knowledge is used, whereas the dictionary size K is evaluated without online adaptation. As illustrated in the figure, accuracy increases as the number of words increases from $K = 10$ to $K = 300$. Figure 6 also gives qualitative results, showing that increasing

3) https://davischallenge.org/index.html.

4) https://youtube-vos.org.

5) DAVIS-2016 set is a subset of DAVIS-2017, containing 30 training videos and 20 validation videos labeled with only a single object.

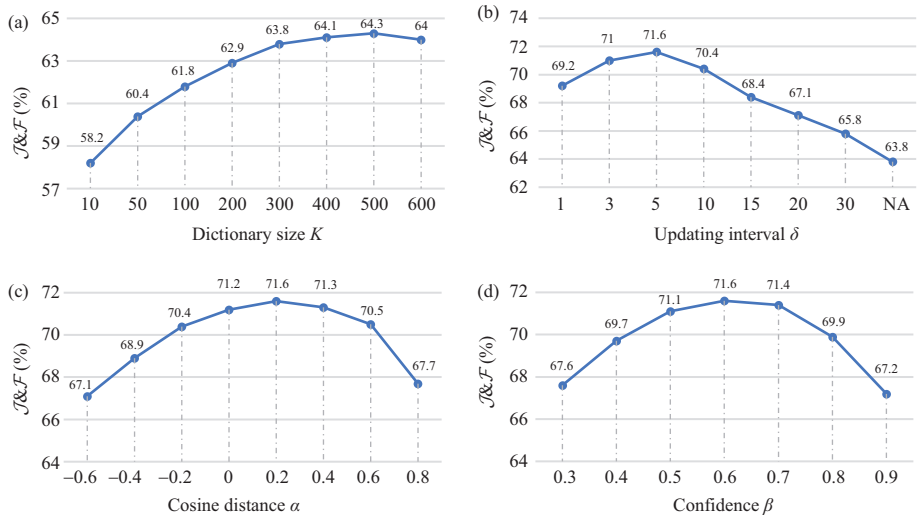

Figure 5 (Color online) Ablative analysis of different choices of hyperparameters on the DAVIS-2017 validation set. (a) Dictionary size K used in Subsection 3.2, (b) updating interval δ, (c) cosine distance α, and (d) confidence β used in Subsection 3.4.

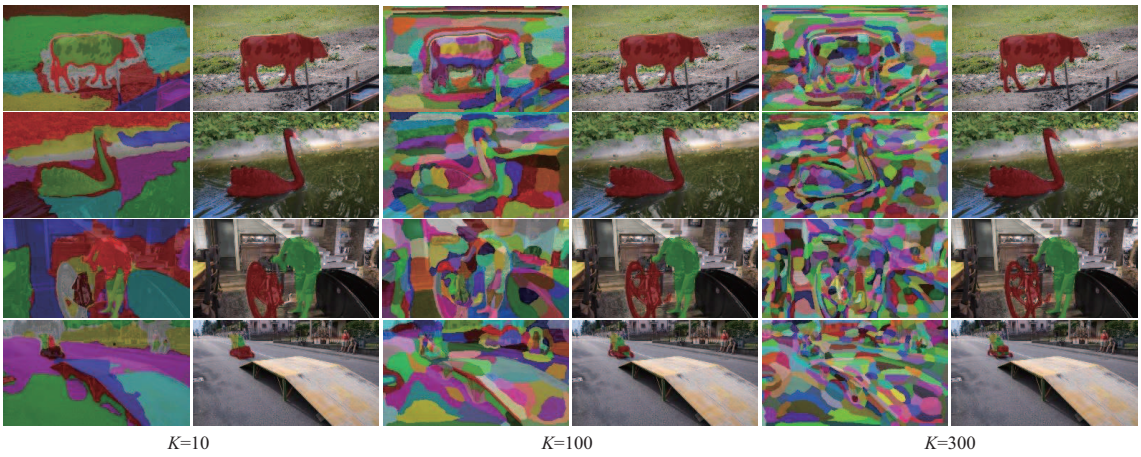

Figure 6 (Color online) Effects of different numbers of visual words at $K = 10$, $K = 100$, and $K = 300$. Different words are represented in different colors (left), which are used to obtain the segmentation output (right).

the number of the visual word dictionary (K) improves the representation of the object and thus improves segmentation outputs. The reason is that it can better capture the intra-object variance. However, the accuracy plateaus between $K = 400$ and $K = 500$ and drops slightly at $K = 600$. We conceive that more words lead to a larger capacity, but this case is prone to overfitting with few-shot information and is time consuming. That is why we set $K = 300$ for our method.

Online adaptation. During online adaptation, three parameters influence performance. A smaller interval implies more frequent updating, which increases the ability of the system to adapt more smoothly to dynamic scenes and outliers. However, excessively small values of δ (e.g., $\delta = 1$) also increase the chance of adding noisy visual words, adversely affecting prediction performance. According to the results, we set $\delta = 5$. Note that the proposed online adaptation has a small computational cost (about 0.4–0.5 s for each updating), as it simply optimizes the target model in a few iterations. The two other parameters are not particularly sensitive in a certain range, but they drop sharply outside this range. That is, it is better not to set them too strict or too loose when updating the module. Through this observation, we set them to their peak values, that is, $\alpha = 0.2$ and $\beta = 0.6$.

4.2 State-of-the-art comparison

Herein, we compare the proposed method with state-of-the-art techniques, including recently developed convolutional neural network-based approaches, on public VOS benchmarks. For a fair comparison, the proposed method is evaluated on DAVIS-2017 [28] by only training the model on the DAVIS-2017 training

Table 2 Quantitative results on DAVIS-2017 validation set

Method	ATD	$\mathcal{J}\&\mathcal{F}$ (%)	$\mathcal{J}$ (%)	$\mathcal{F}$ (%)	Speed (s)
GC [20]	IM	71.4	69.3	73.5	0.08*
FEELVOS [15]	YT	71.5	69.1	74.0	0.51
AGAME [12]	IM+YT	71.0	68.5	73.6	0.07
AGSS [16]	YT	67.4	64.9	69.9	0.10
FRTM [41]	YT	76.7	–	–	0.05
AFB-URR [21]	IM	74.5	73.0	76.1	0.18
STM [19]	IM	71.6	69.2	74.0	0.32*
STM [19]	IM+YT	**81.7**	**79.2**	**84.3**	0.32*
Ours	YT	73.4	70.1	76.7	0.13
OSVOS-S [8]	–	68.0	64.7	71.3	9.0*
MLDVW [43]	–	67.3	63.9	70.7	0.29
FEELVOS [15]	–	69.1	65.9	72.3	0.51
AGAME [12]	–	63.2	–	–	0.07
AGSS [16]	–	66.6	63.4	69.8	0.10
FRTM [41]	–	68.8	–	–	0.05
STM [19]	–	43.0	38.1	47.9	0.32*
Ours (fast)	–	63.8	60.7	66.9	0.04
Ours	–	**71.6**	**68.5**	**74.8**	0.13

a) ATD: additional training datasets used in the respective method. Specifically, 'IM' indicates static image datasets, and 'YT' denotes YouTubeVOS-2018 training dataset. '$*$': Timing extrapolated from DAVIS-2016 assuming linear scaling in the number of objects. Ours (fast): our approach without online adaptation.

set and is evaluated on YouTubeVOS-2018 [29] by only training the model on the YouTubeVOS-2018 training set. We also provide the results using both sets, following some latest studies. To highlight the strengths of the proposed architecture, we indicate whether an additional training set is used for each compared method.

DAVIS-2017. DAVIS-2017 [28] contains 90 full high-definition videos densely annotated with pixel-level accurate object masks in all frames. Among them, 60 videos are used for training and 30 videos are used for validation. Each video includes one or multiple foreground objects. Table 2 summarizes the results. All the records in the table and the per-frame runtime are available in the original papers. As runtime is not reported in some studies, we time them by extrapolating from the result on DAVIS-2016, assuming linear scaling in the number of objects. Some methods [7, 8, 30] rely heavily on online training to finetune a network in the first frame. This type of training usually requires several minutes of GPU-powered configuration for each test video. Despite their high accuracy, these methods have high computational costs. To allow a fair comparison, these online steps are included in the runtime calculation by averaging over all frames. By contrast, without time-consuming finetuning, the recent methods require less than 0.5 s to process each frame.

Moreover, we find that most of the compared methods improve the accuracy by using additional datasets. Among previous approaches, STM [19] obtains the highest overall $\mathcal{J}\&\mathcal{F}$ (%) score by pre-training their model on large-scale images and YouTubeVOS data. However, the performance of STM is notably reduced to 43.0 without the use of additional training data. Other methods also have similar performances. By contrast, our approach obtains the best overall score of 71.6% when using only the DAVIS-2017 training set. It clearly demonstrates the strength of our target model. Given that the proposed model learns the target without backpropagation, our system requires no pretraining on large-scale static images, whereas most methods [10, 12, 19–21] need. Although the proposed method requires a lightweight learning step that predicts the target model, it is not excessively time consuming because of the employed fast optimization techniques. Figure 7 gives an illustration in terms of speed/accuracy tradeoffs, where FRTM [41] has been the fastest method so far. We also provide the results using an additional YouTubeVOS set. Interestingly, our method is not that sensitive to the used training set, where the overall score rises from 71.6% to 73.4%. The reason is that the proposed target model contributes more than the learned meta knowledge for improving the accuracy, which can also be inferred from REH and RTM in Table 1.

YouTubeVOS-2018. YouTubeVOS-2018 [29] is the latest large-scale dataset for the VOS that comprises 3471 videos for training and 474 videos for validation. In addition, the validation set has 26 unseen categories, which do not exist in the training dataset, to evaluate the generalization ability of

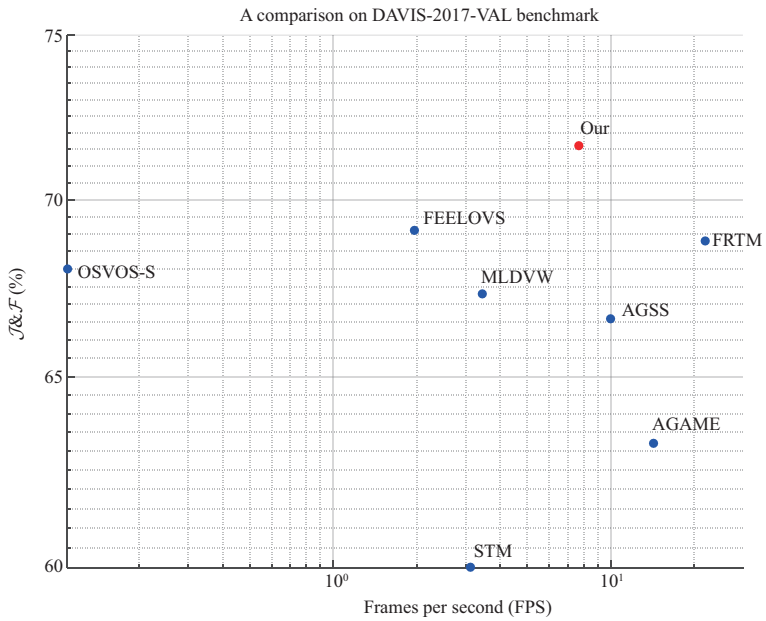

Figure 7 (Color online) Comparison of accuracy and speed when only using DAVIS-2017 for training.

Table 3 Quantitative results on YouTubeVOS-2018 validation set[a)]

Method	ATD	$\mathcal{J}\&\mathcal{F}$ (%)	$\mathcal{J}$ (%)		$\mathcal{F}$ (%)	
		Overall	Seen	Unseen	Seen	Unseen
PReMVOS [7]	IM	66.9	71.4	56.5	–	–
GC [20]	IM	73.2	72.6	68.9	75.6	75.7
AFB-URR [21]	IM	**79.6**	78.8	74.1	**83.1**	**82.6**
STM [19]	IM	79.4	**79.7**	**84.2**	72.8	80.9
Ours	DV	76.1	73.6	72.2	77.6	81.0
STM [19]	–	68.2	–	–	–	–
AGAME [12]	–	66.1	67.8	61.2	69.5	66.2
AGSS [16]	–	71.3	71.3	65.5	75.2	73.1
FRTM [41]	–	72.1	72.3	65.9	76.2	74.1
Ours	–	**75.4**	**73.5**	**71.6**	**77.4**	**80.3**

a) 'IM' indicates static image datasets. 'DV' denotes DAVIS-2017 dataset.

algorithms. Based on the results presented in Table 3, our approach obtains the best overall score of 75.4% when using only the YouTubeVOS-2018 training set. Even compared with the methods pretrained on large-scale images, the proposed approach remains competitive with these state-of-the-art algorithms in terms of speed/accuracy tradeoffs. Furthermore, we find that our method significantly outperforms other methods [12, 16, 19, 41] in unseen categories without additional datasets. This result validates that our model can learn a new object well from few-shot information in a fast way.

4.3 Qualitative evaluation

Figure 8 presents qualitative comparison results on DAVIS-2017, and the compared methods here are all only trained without additional datasets. The results show that the proposed method produces better results than others in most cases. Specifically, the occluded man, the newly exposed motorbike, and the appearance-changing boy and girl can be easily captured by our system because of the powerful learned target model. However, we find that it struggles to split very similar objects (e.g., the dancing boy and the audience behind him) when they are near. The reason is that the instances have a similar appearance and are close to one another, resulting in excessively proximate embeddings. The limitation can be more or less addressed by local matching techniques, such as FEELVOS [15], yet it suffers from error accumulation.

More qualitative evaluations of the proposed approach on the DAVIS-2017 and YouTubeVOS-2018 are displayed in Figure 9. The method is capable of producing satisfactory results in quite challenging

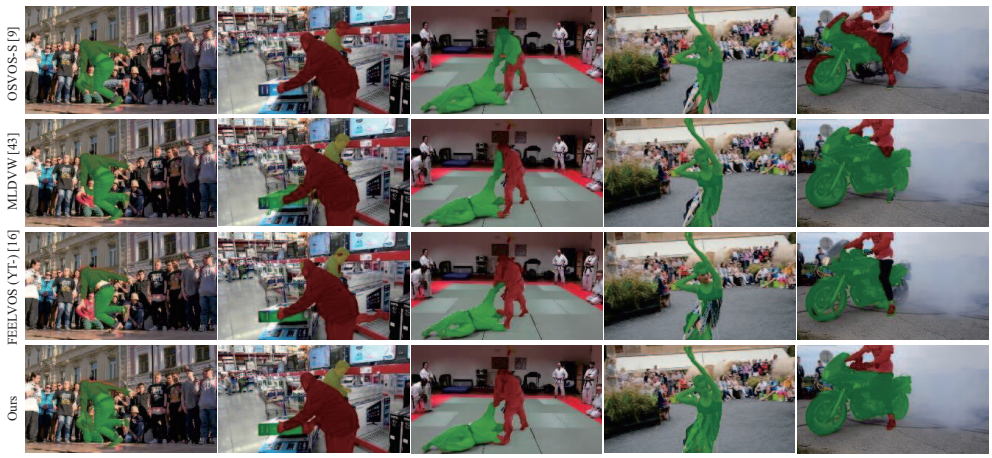

Figure 8 (Color online) Qualitative comparison results on DAVIS-2017, where the compared methods are all only trained on DAVIS-2017 without additional datasets. The proposed method produces better results than others in most cases. However, it struggles to split similar objects when they are close to one another.

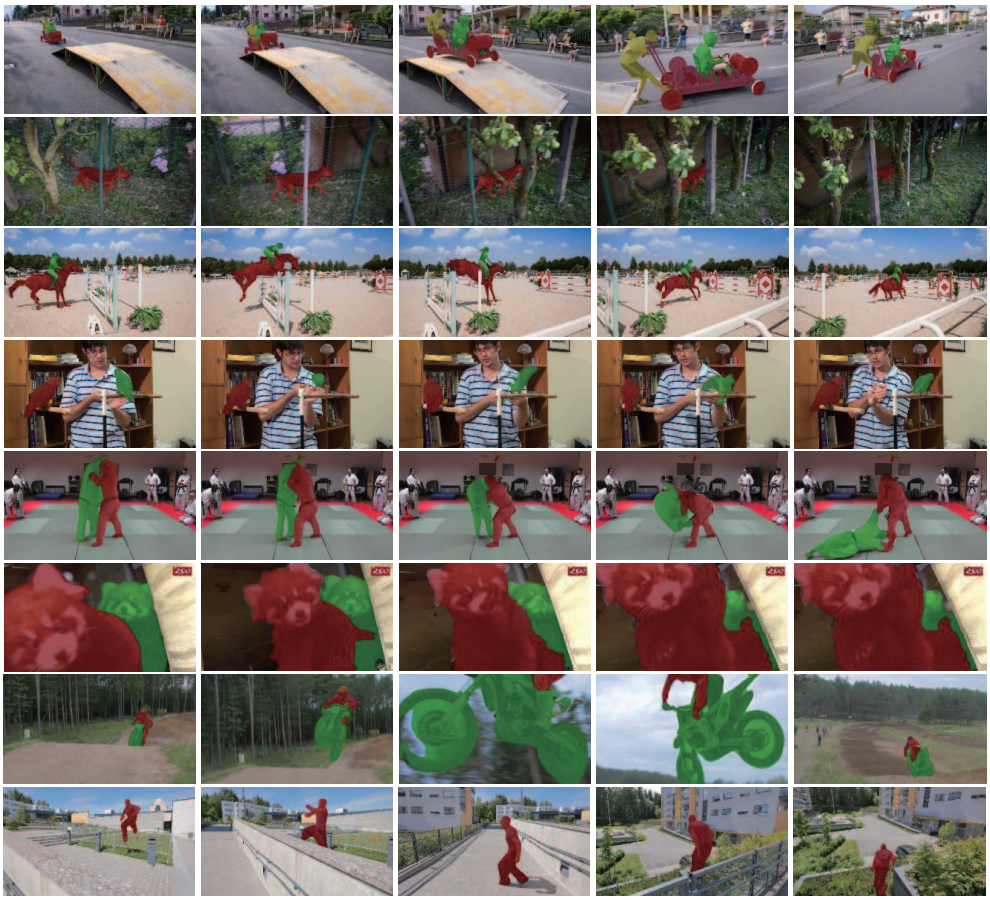

Figure 9 (Color online) More qualitative results of our VOS method on DAVIS-2017 and YouTubeVOS-2018 datasets. Our method can produce good results even in challenging scenarios, including new exposures (rows 1, 7, 8), occlusions (rows 2, 3), appearance changes (rows 1, 7), fast motions (rows 3, 4, 8), very similar objects (rows 4, 5, 6).

situations, such as occlusions, appearance changes, fast motions, and object similarities. Even though the full object appearance is not revealed in the first frame (rows 1, 7, 8), the proposed method successfully captures the target information through few-shot learning. Inevitably, however, the method cannot fully capture some detailed parts, such as the human legs (denoted by yellow in row 1) or the foot of the rider and the back seat of the motorbike (row 7). A possible reason is that the missed information is undetected on the object in the first frame, but it is highly similar to that of other distractor objects.

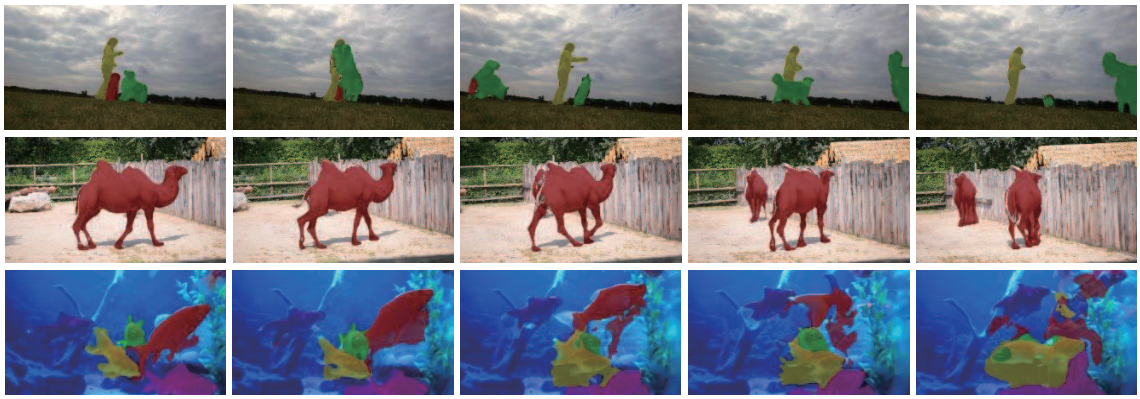

Figure 10 (Color online) Some failure cases due to overlapped distractor objects (rows 1, 2, 3) and severe blur (row 3). The reason is that the instances have a similar appearance and are close to one another, resulting in excessively proximate embeddings.

As aforementioned, our method struggles to split very similar objects when they intertwine together (rows 5, 6). This limitation can also be seen in Figure 10. For example, in rows 1 and 2 of Figure 10, the different dogs or camels are predicted as the same instance. In the challenging fish sequence (row 3), parts of objects are also lost, and distinguishing fish is difficult. Although we adapt the target model online, distinguishing them using similar features in embedding space is difficult. Moreover, in row 3 of Figure 10, the fins have a blurry appearance, owing to their transparent appearance and fast motion.

5 Conclusion

We develop a fast-learning approach for few-shot VOS, where the meta knowledge learned from various tasks is used to train a target model for a new task. In this approach, meta knowledge is represented as backbone extraction with an embedding head, and the target-aware model comprises generative and discriminative learning procedures. The model is trained by the bilevel-optimization strategy across several different tasks sampled from the public VOS training dataset. Owing to the powerful target model and the easy training strategy, the proposed method is simple, fast, target-aware, strong, and robust. Moreover, it can segment multiple objects per video in a single forward pass. Without using additional training data, time-consuming finetuning, optical flows, or pre/postprocessing, the proposed method compares favorably with state-of-the-art methods on DAVIS-2017, with a $\mathcal{J}\&\mathcal{F}$ overall score of 71.6%, and on YouTubeVOS-2018, with a $\mathcal{J}\&\mathcal{F}$ overall score of 75.4%; furthermore, it maintains a high inference speed of approximately 0.13 s per frame. By training the method on DAVIS-2017 and YouTubeVOS-2018, accuracy can be further increased to 73.4% and 76.1%, respectively.

Acknowledgements This work was partially supported by National Natural Science Foundation of China (Grant Nos. 62072449, 61802197), Science and Technology Development Fund, Macao SAR (Grant Nos. 0018/2019/AKP, SKL-IOTSC(UM)-2021-2023), Guangdong Science and Technology Department (Grant No. 2018B030324002), and Zhuhai Science and Technology Innovation Bureau Zhuhai-Hong Kong-Macau Special Cooperation Project (Grant No. ZH22017002200001PWC).

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Wu W M, Wang Q, Yuan C Z, et al. Rapid dynamical pattern recognition for sampling sequences. Sci China Inf Sci, 2021, 64: 132201
2 Gu Y F, Liu H, Wang T F, et al. Deep feature extraction and motion representation for satellite video scene classification. Sci China Inf Sci, 2020, 63: 140307
3 Chen Y D, Hao C Y, Wu W, et al. Robust dense reconstruction by range merging based on confidence estimation. Sci China Inf Sci, 2016, 59: 092103
4 Perazzi F, Khoreva A, Benenson R, et al. Learning video object segmentation from static images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017
5 Caelles S, Maninis K K, Pont-Tuset J, et al. One-shot video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 5320–5329
6 Lu X K, Wang W G, Shen J B, et al. Learning video object segmentation from unlabeled videos. In: Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition, 2020. 8957–8967

7 Luiten J, Voigtlaender P, Leibe B. PReMVOS: proposal-generation, refinement and merging for video object segmentation. In: Proceedings of the 2018 DAVIS Challenge on Video Object Segmentation—CVPR Workshops, 2018

8 Maninis K K, Caelles S, Chen Y, et al. Video object segmentation without temporal information. IEEE Trans Pattern Anal Mach Intell, 2019, 41: 1515–1530

9 Khoreva A, Benenson R, Ilg E, et al. Lucid data dreaming for video object segmentation. Int J Comput Vis, 2019, 127: 1175–1197

10 Oh S W, Lee J, Sunkavalli K, et al. Fast video object segmentation by reference-guided mask propagation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 7376–7385

11 Xiao H, Feng J, Lin G, et al. MoNet: deep motion exploitation for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 1140–1148

12 Johnander J, Danelljan M, Brissman E, et al. A generative appearance model for end-to-end video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 8945–8954

13 Xie H Z, Yao H X, Zhou S C, et al. Efficient regional memory network for video object segmentation. 2021. ArXiv:2103.12934

14 Hu Y T, Huang J B, Schwing A G. VideoMatch: matching based video object segmentation. In: Proceedings of the 2018 European Conference on Computer Vision, 2018

15 Voigtlaender P, Chai Y, Schroff F, et al. FEELVOS: fast end-to-end embedding learning for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 9473–9482

16 Lin H, Qi X, Jia J. AGSS-VOS: attention guided single-shot video object segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 3948–3956

17 Yang Z X, Wei Y C, Yang Y. Collaborative video object segmentation by foreground-background integration. In: Proceedings of the European Conference on Computer Vision, 2020

18 Vaswani A, Shazeera N, Parmar N, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017. 6000–6010

19 Oh S W, Lee J, Xu N, et al. Video object segmentation using space-time memory networks. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 9225–9234

20 Li Y, Shen Z R, Shan Y. Fast video object segmentation using the global context module. In: Proceedings of the European Conference on Computer Vision, 2020

21 Liang Y Q, Li X, Jafari N, et al. Video object segmentation with adaptive feature bank and uncertain-region refinement. In: Proceedings of the Conference on Neural Information Processing Systems, 2020

22 Wang H C, Jiang X L, Ren H B, et al. SwiftNet: real-time video object segmentation. 2021. ArXiv:2102.04604

23 Hu L, Zhang P, Zhang B, et al. Learning position and target consistency for memory-based video object segmentation. 2021. ArXiv:2104.04329

24 Duke B, Ahmed A, Wolf C, et al. SSTVOS: sparse spatiotemporal transformers for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021

25 Chen Y D, Hao C Y, Liu A X, et al. Multilevel model for video object segmentation based on supervision optimization. IEEE Trans Multimedia, 2019, 21: 1934–1945

26 Hao C Y, Chen Y D, Yang Z X, et al. Higher-order potentials for video object segmentation in bilateral space. Neurocomputing, 2020, 401: 28–35

27 Chen Y D, Hao C Y, Liu A X, et al. Appearance-consistent video object segmentation based on a multinomial event model. ACM Trans Multimedia Comput Commun Appl, 2019, 15: 1–15

28 Pont-Tuset J, Perazzi F, Caelles S, et al. The 2017 DAVIS challenge on video object segmentation. 2017. ArXiv:1704.00675

29 Xu N, Yang L J, Fan Y C, et al. YouTube-VOS: a large-scale video object segmentation benchmark. 2018. ArXiv:1809.03327

30 Voigtlaender P, Leibe B. Online adaptation of convolutional neural networks for video object segmentation. In: Proceedings of the British Machine Vision Conference, 2017

31 Li X X, Loy C C. Video object segmentation with joint re-identification and attention-aware mask propagation. In: Proceedings of the European Conference on Computer Vision, 2018

32 Griffin B A, Corso J J. BubbleNets: learning to select the guidance frame in video object segmentation by deep sorting frames. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 8906–8915

33 Tian Z, He T, Shen C. Decoders matter for semantic segmentation: data-dependent decoding enables flexible feature aggregation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 3121–3130

34 Bao L C, Wu B Y, Liu W. CNN in MRF: video object segmentation via inference in a CNN-based higher-order spatio-temporal MRF. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018

35 Zhang Y, Wu Z, Peng H, et al. A transductive approach for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 6947–6956

36 Zhang K H, Wang L, Liu D, et al. Dual temporal memory network for efficient video object segmentation. 2020. ArXiv:2003.06125

37 Chen Y, Pont-Tuset J, Montes A, et al. Blazingly fast video object segmentation with pixel-wise metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 1189–1198

38 Hospedales T, Antoniou A, Micaelli P, et al. Meta-learning in neural networks: a survey. 2020. ArXiv:2004.05439

39 Yang L, Wang Y, Xiong X, et al. Efficient video object segmentation via network modulation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 6499–6507

40 Tang L L, Chen K, Wu C, et al. Improving semantic analysis on point clouds via auxiliary supervision of local geometric priors. IEEE Trans Cybern, 2020, 12: 1–11
41 Robinson A, Lawin A J, Danelljan M, et al. Learning fast and robust target models for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 7404–7413
42 Bhat G, Lawin F G, Danelljan M, et al. Learning what to learn for video object segmentation. In: Proceedings of the European Conference on Computer Vision, 2020
43 Behl H S, Najafi M, Arnab A, et al. Meta learning deep visual words for fast video object segmentation. In: Proceedings of the Conference on Neural Information Processing Systems Machine Learning for Autonomous Driving Workshop, 2019
44 Pinheiro P, Lin T Y, Collobert R, et al. Learning to refine object segments. In: Proceedings of the European Conference on Computer Vision, 2016. 75–91
45 Chen L C, Papandreou G, Kokkinos I, et al. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell, 2018, 40: 834–848
46 He K M, Zhang X, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778
47 Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the Machine Learning Research, 2017. 1126–1135
48 He K M, Zhang X, Ren S Q, et al. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision, 2015. 1026–1034